1. You have been asked to analyse some data which is in a text file. The data is formatted in blocks of varying sizes, with a header for each block. Consulting your notes, you discover you need the function `readLines` which you have never used. How would you find out how to use it?

2. (a) Create a character vector called `filmNames` containing the names of 4 films (imaginary or genuine!).

   (b) Create a numeric vector called `filmTimes` containing the estimated running times in minutes for these films.

   (c) Create a list of length 2 called `filmsList` consisting of the two elements: `filmNames` and `filmTimes`.

   (d) Create a vector called `goodFilms` containing the logical values TRUE, FALSE, TRUE, TRUE.

   (e) How could you extract a vector from `filmsList` containing those film names which correspond to TRUE values in `goodFilms`?

   (f) Use **lapply** to extract the names and running times for those films which correspond to FALSE values in `goodFilms`. What will be the format of the result?

3. Write a function which contains a "for" loop and for any given **n** acquires **n** uniform random numbers and then calculates their mean. How in practice would you do this in R?

4. Suppose `betas` is a n by p matrix containing some results from a Bayesian MCMC model fitting: values of the parameters $(\beta_i, i = 1, \ldots, p)$ for each of n steps of the chain. Write the code to do the following:

   (a) Use **apply** to calculate the acf for each column of the matrix. (Assume the function `acf(x)`, if called with a vector x, will return a list which has a component named `acf` which is an array of dimensions `c(maxlag + 1, 1, 1)` containing the auto-correlations at lags from 0 upwards for some maximum lag `maxlag`. It will also produce a plot: to suppress this you should use the argument `plot=FALSE`.)

   (b) Find the maximum absolute value among the columns of auto-correlation at lag 1.

   (c) Extract a sub matrix containing every 10th row.

   (d) Write a function to iterate this: calculate the maximum absolute value of the auto-correlation at lag 1 and then thin by extracting every 10th row until the maximum absolute value of the auto-correlation at lag 1 is less than $p/\sqrt(m)$, where m is the number of rows in the sub matrix, for some specified p, and then return the thinned matrix.

   You may ignore the possibility of input error, and assume that p is large enough for the process to stop before you run out of rows.

   Some credit will be given for pseudo-code mixed with R.

answers

1. 1 mark

```
?readLines
```

2. (a) 1/2 mark

   ```
   filmNames <- c("Local Hero", "Gone With the Wind",
          "Psycho", "Brief Encounter")
   ```

   (b) 1/2 mark

   ```
   filmTimes <- c(95, 130, 100, 80)
   ```

   (c) 1/2 mark

   ```
   filmsList <-  list(filmNames, filnTimes)
   ```

   (d) 1/2 mark

   ```
   goodFilms <- c(TRUE, FALSE, TRUE, TRUE)
   ```

   (e) 1 mark

   ```
   filmsList[[1]][goodFilms] ## note [[
   ```

   (f) 1 mark

   ```
   lapply(filmsList, function(x) x[!goodFilms])
   ```

   Result will be a list

3. 5 marks

```
myfn <- function(n)
{
  store <- 0
  x <- runif(n)
  for (i in 1:n)
  {
    store <- store + x[i]
  }
  store/n
}
n <- 100
mean(runif(n))
```

4. (a) 2 marks

   ```
   myacf <- apply(betas, 2, acf, plot=FALSE)
   ```

   (b) 2 mark

   ```
   myacfval <- sapply(myacf, function(x)x$acf[2, 1, 1])
   mymax <- max(myacfval)
   ```

   (c) 1 mark

   ```
   betas <- betas[c(rep(FALSE, 9), TRUE), ]
   ```

   (d) 5 marks

   ```
   myfn <- function(betas, p)
   {
        repeat
        {
   ```

```
        myacf <- apply(betas,2, acf, plot=FALSE)
        myacfval <- sapply(myacf, function(x)x$acf[2, 1, 1])
        mymax <- max(abs(myacfval))
        if (mymax < p/sqrt(nrow(betas)))
        {
           break
        }
        betas <- betas[c(rep(FALSE, 9), TRUE), ]
    }
    betas
}
```